Consolidated Run-Time Infrastructure Functional Capabilities

Presented to

Architecture Management Group

Alexandria, VA
9 August 1995

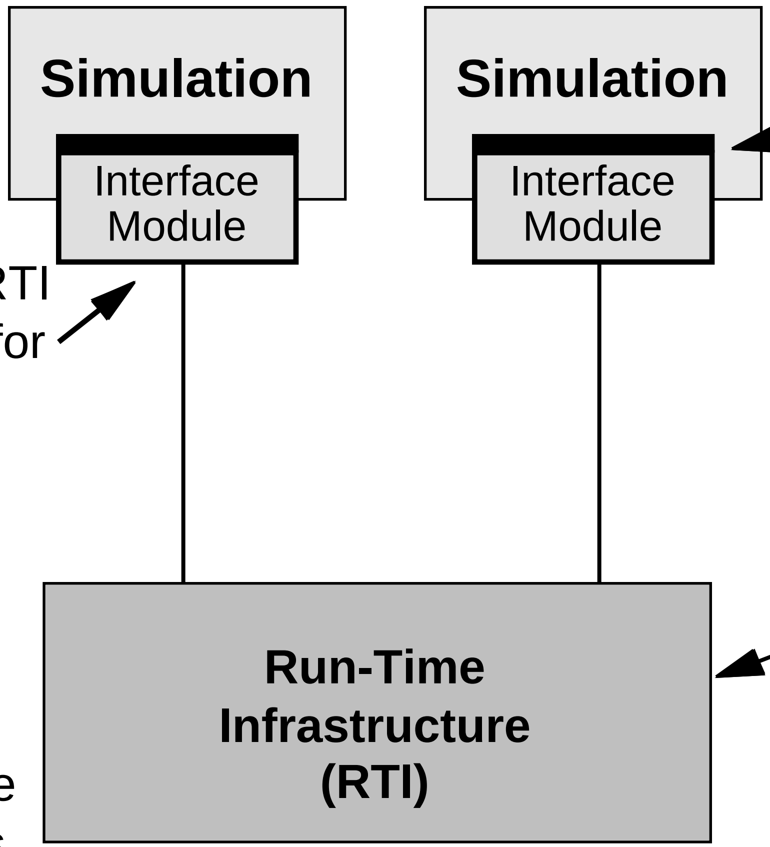
Duncan C. Miller, Sc.D.

M.I.T. Lincoln Laboratory
244 Wood St.
Lexington, MA 02173-9108
617-981-7452

dmiller@ll.mit.edu

Consolidated RTI Conceptual Overview

Some DIS 3.x functions
will be required
in each simulation



Application Program
Interface (API), based on
HLA Interface Spec,
Available in C and Ada

The interface to the RTI
will be standardized for
all federations and
executions

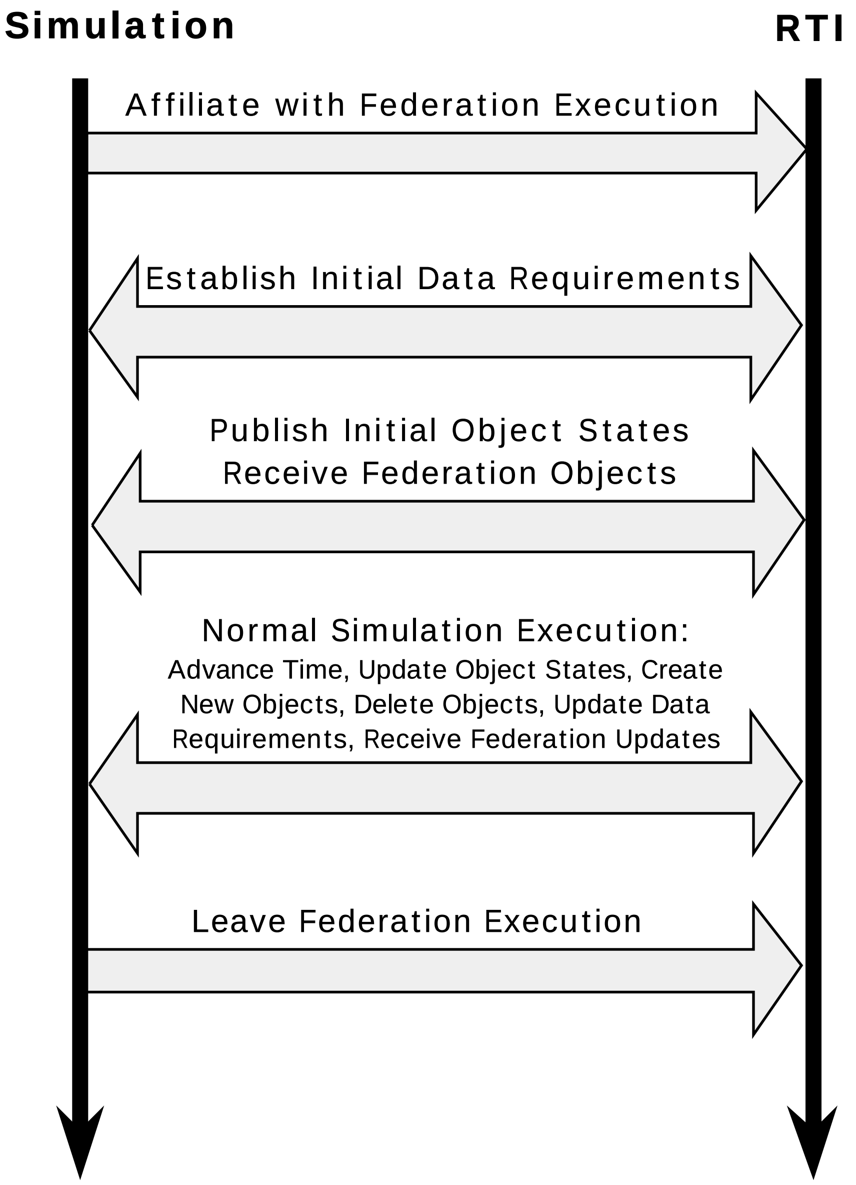
But ... not all RTI
functions are used
by all federations,
and different
federations may use
alternative versions
of some functions

Simulations don’t need
to be concerned with
where or how the RTI
services are provided

RTI Service Groups

- Federation Management
- Declaration Management
- Object Management
- Ownership Management
- Time Management

Overall Simulation/RTI Interactions

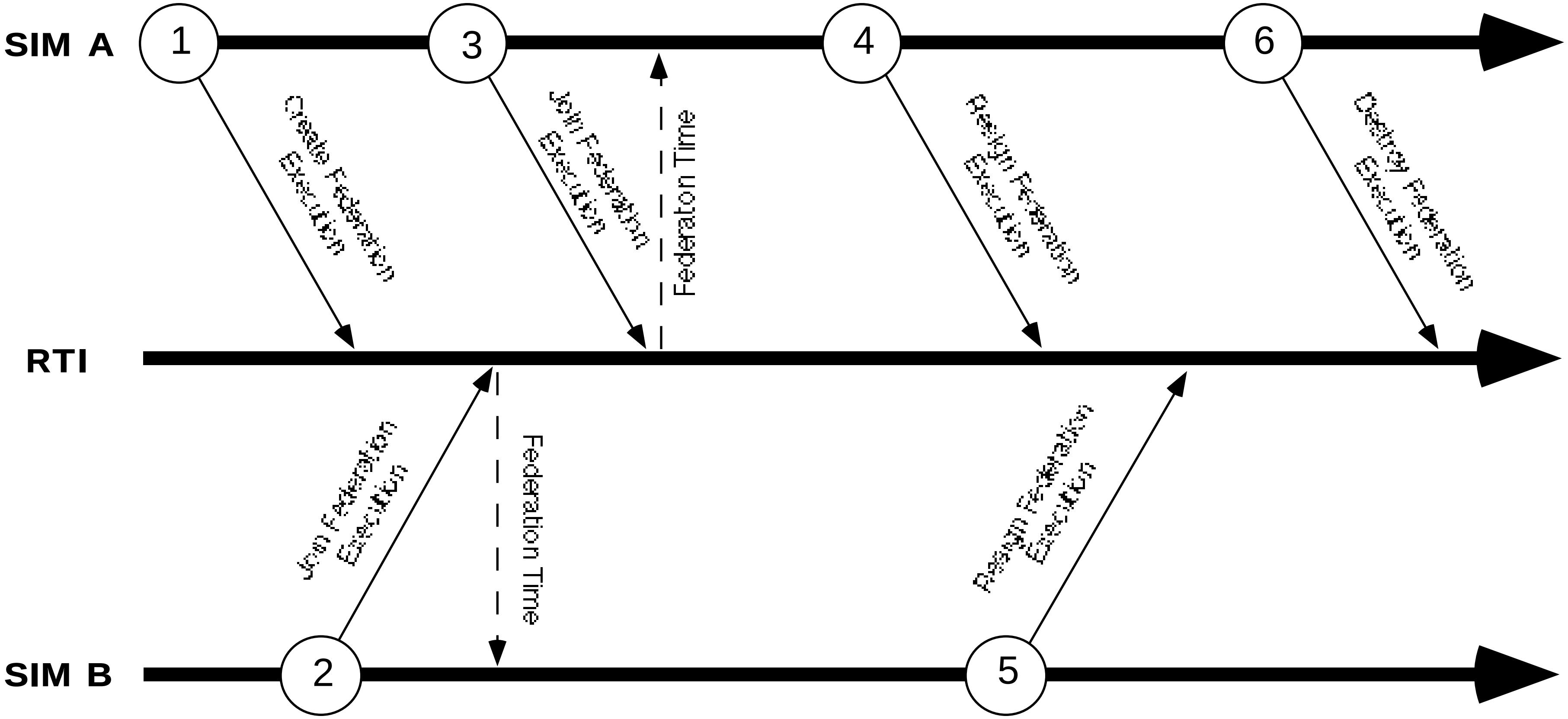

Simulation
RTI
Affiliate with Federation Execution
Establish Initial Data Requirements
Publish Initial Object States
Receive Federation Objects
Normal Simulation Execution:
Advance Time, Update Object States, Create New Objects, Delete Objects, Update Data Requirements, Receive Federation Updates
Leave Federation Execution

Federation Management Services

Service	Invoked by
Create Federation Execution	Simulation
Destroy Federation Execution	Simulation
Join Federation Execution	Simulation
Resign Federation Execution	Simulation
Request Pause	Simulation
Pause	RTI
Pause Achieved	Simulation
Request Resume	Simulation
Resume	RTI
Resume Achieved	Simulation
Schedule Federation Save	Simulation
Start Federation Save	RTI
Federation Save Begun	Simulation
Federation Save Complete	Simulation
Restore	RTI
Restore Complete	Simulation
Submit Query	Simulation
Query	RTI
Query Result	RTI

Federation Management Services

Federation Creation


SIM A
1
3
4
6
Create Federation Execution
Join Federation Execution
Federaton Time
Resign Federation Execution
Destroy Federation Execution
RTI
Join Federation Execution
Federation Time
Resign Federation Execution
SIM B
2
5

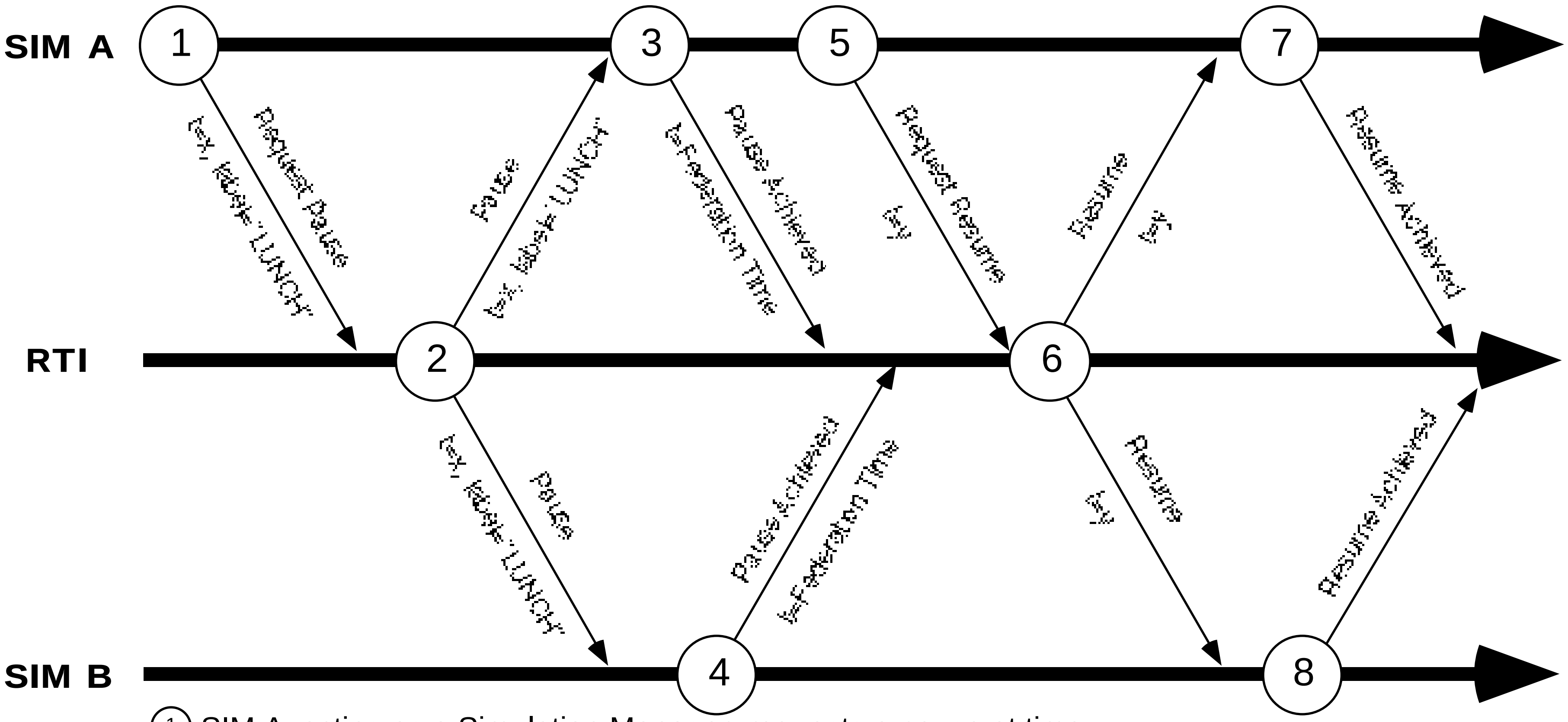
1. SIM A, acting as a Simulation Manager, creates the Federation Execution
2. SIM B joins the Federation Execution and receives the current Federation Time
3. SIM A joins the Federation Execution and receives the current Federation Time

4 & 5 Sim A and SIM B resign from the Federation Execution

6. After all simulations have resigned, SIM A tells the RTI to destroy the Federation Execution

Federation Management Services

Federation Pause and Resume



1 SIM A, acting as a Simulation Manager, requests a pause at time x

2 The RTI issues the Pause notification to the simulations in the Federation Execution

3 & 4 When time x arrives, SIM A and SIM B pause and respond that the pause has been achieved, noting the time

5 Sim A requests a resume at time = y

6 The RTI issues a Resume notification to the simulations in the Federation Execution

7 & 8 After time = y, SIM A and SIM B resume and respond that the Resume has been achieved

NOTE: x = federation time, y = wall clock time

Duncan Miller
4/15/96

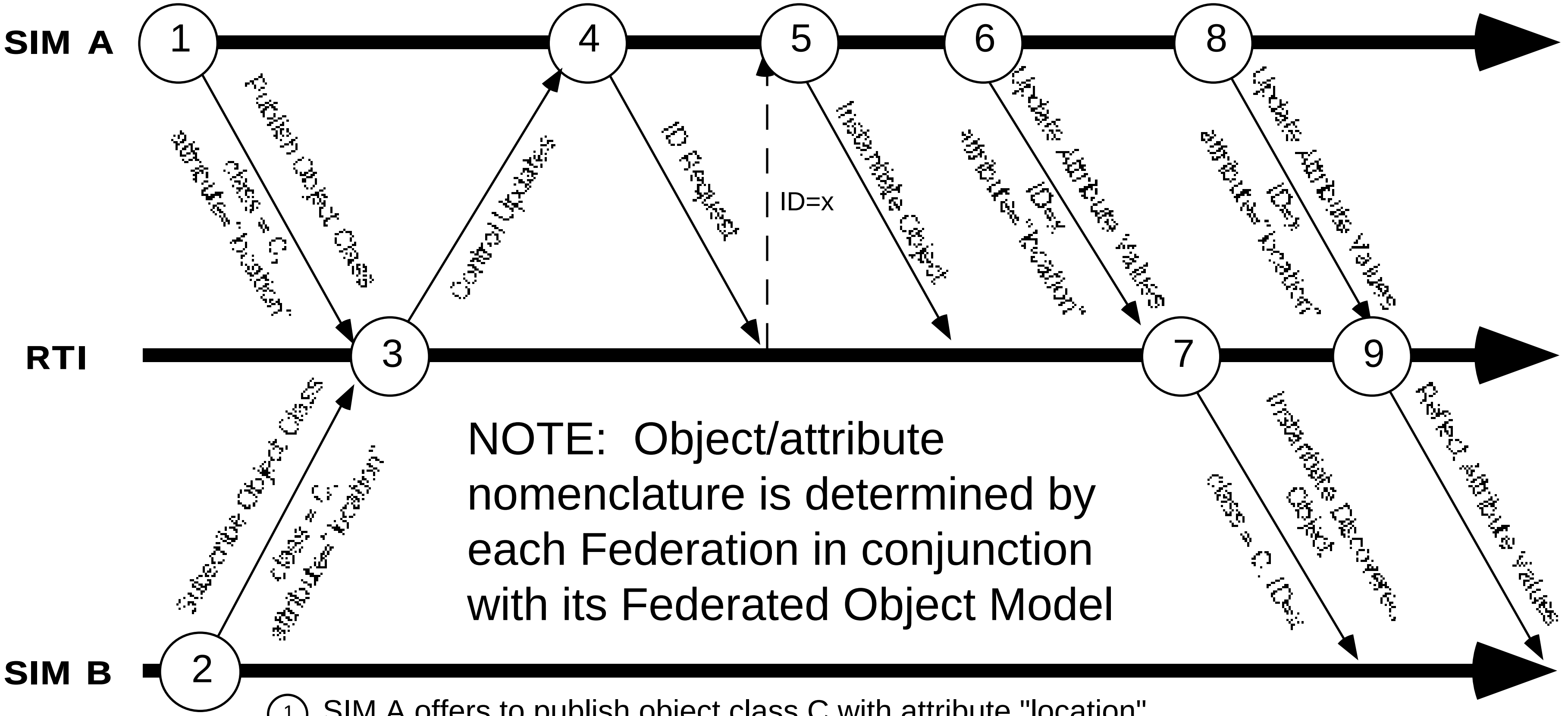
Declaration Management Services

Service	Invoked by
Publish Object Class	Simulation
Publish Object Attributes	Simulation
Publish Interaction Class	Simulation
Subscribe Object Class	Simulation
Subscribe Object Attributes	Simulation
Subscribe Interaction Class	Simulation
Control Updates	RTI
Control Interactions	RTI

Object Management Services

Service	Invoked by
ID Request	Simulation
Instantiate Object	Simulation
Instantiate Discovered Object	RTI
Delete Object	Simulation
Remove Object	RTI
Update Attribute Values	Simulation
Reflect Attribute Values	RTI
Cancel Object Attribute Updates	Simulation
Send Interaction	Simulation
Receive Interaction	RTI
Obtain Attribute Values	Simulation
Provide Attribute Values	RTI

Declaration Management Services and Object Management Services



NOTE: Object/attribute nomenclature is determined by each Federation in conjunction with its Federated Object Model

1. SIM A offers to publish object class C with attribute "location"
2. SIM B Subscribes to class C and attribute "location"
3. The RTI informs SIM A that updates will be required for objects of class C
4. SIM A requests an ID for an object in class C it intends to instantiate; the RTI provides ID = x
5. SIM A instantiates the object

6 & 7 SIM A updates the attributes of object x and the RTI instantiates the discovered object for all subscribing simulators whose discovery predicates are true

8 & 9 The RTI forwards updates of the object to subscribing simulators

Duncan Miller
4/15/96

Time Management Services

Service	Invoked by
Set Federation Time	Simulation
Request Federation Time	Simulation
Set Federation Rate	Simulation
Time Advance Request	Simulation
Time Advance Grant	RTI

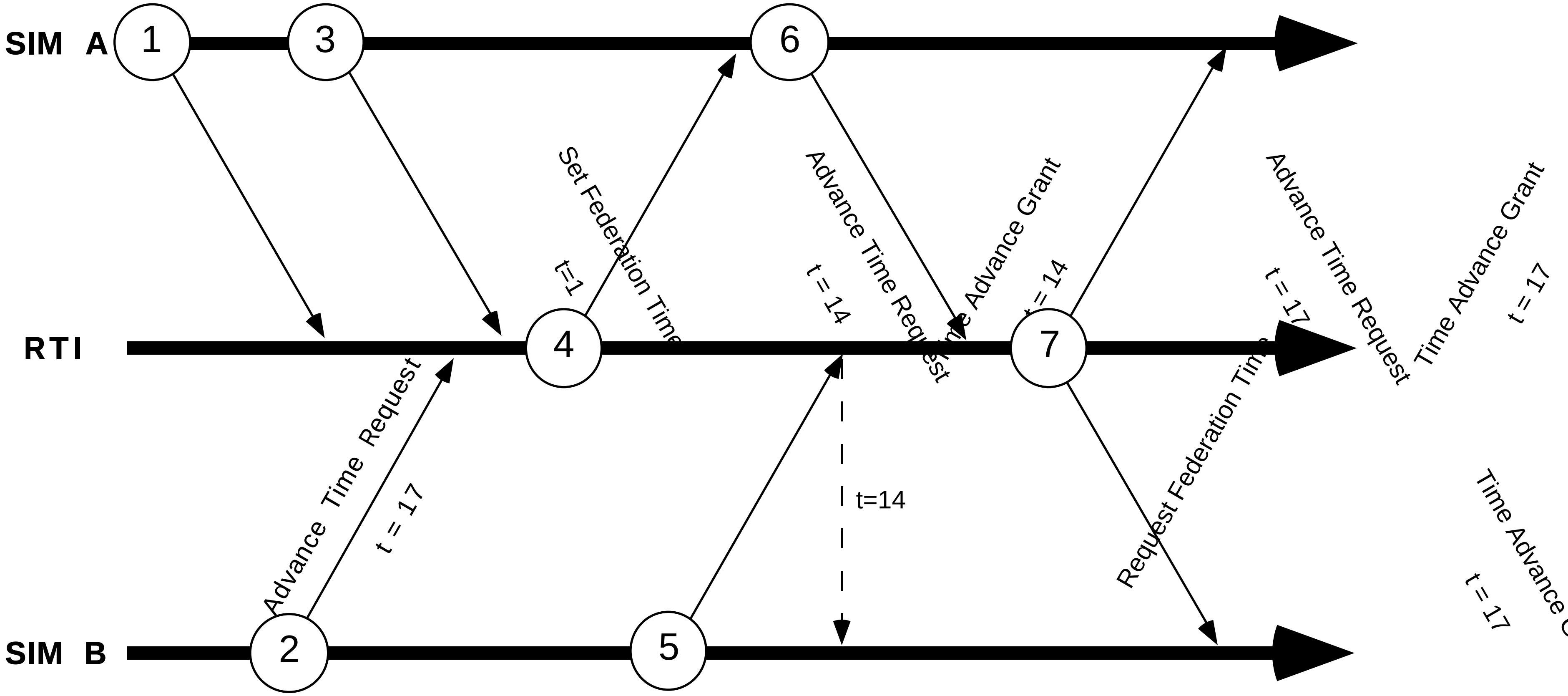
Definitions:

Paced — Federation time proceeds at a constant rate with a fixed ratio to real “wallclock” time

Agreement — Federation time does not advance without explicit consent from the RTI

Time Management Services

Not Paced, With Agreement



1 SIM A sets the initial federation time (to t = 1)

2 & 3 SIM B requests time to advance to t=17, SIM A requests time to advance to t=14

4 The RTI allows SIM A to advance to to t=14, but does not grant the request to SIM B

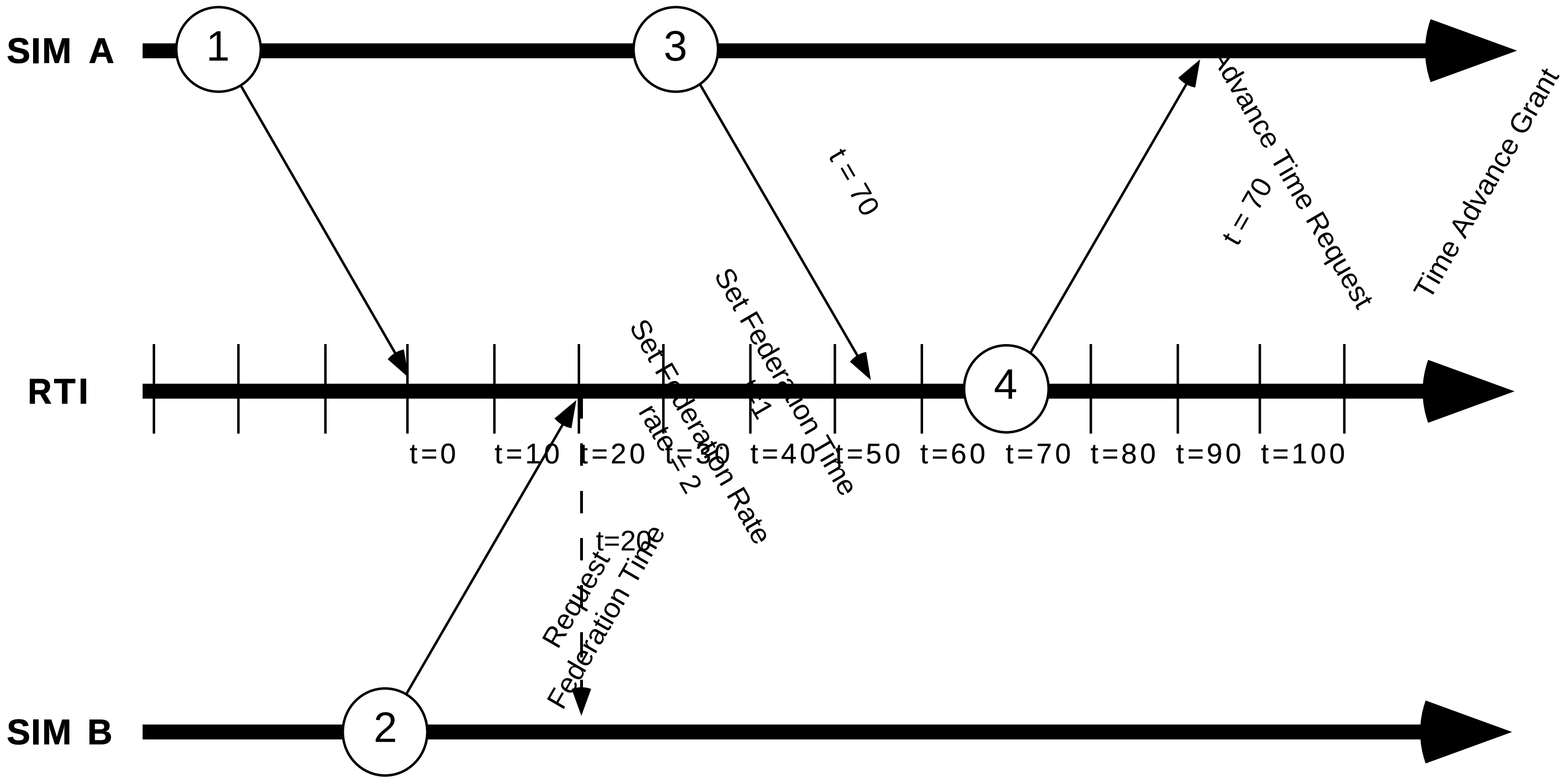
5 SIM B requests the current Federation time (out of curiosity). The Federation is at t=14.

6 SIM A now requests time to advance to t=17

7 The RTI allows both simulators to advance to t=17 since they have both requested it.

Time Management Services

Paced With No Agreement



① SIM A, acting as Simulation Manager, sets the current federation time and rate

② After Joining the Federation, SIM B requests the current federation time and the RTI responds

③ & ④ SIM A uses the Advance/Grant mechanism to receive a notification of t = 70.

NOTE: In this federation, each simulator maintains its own clock and would request federation time whenever synchronization is desired.

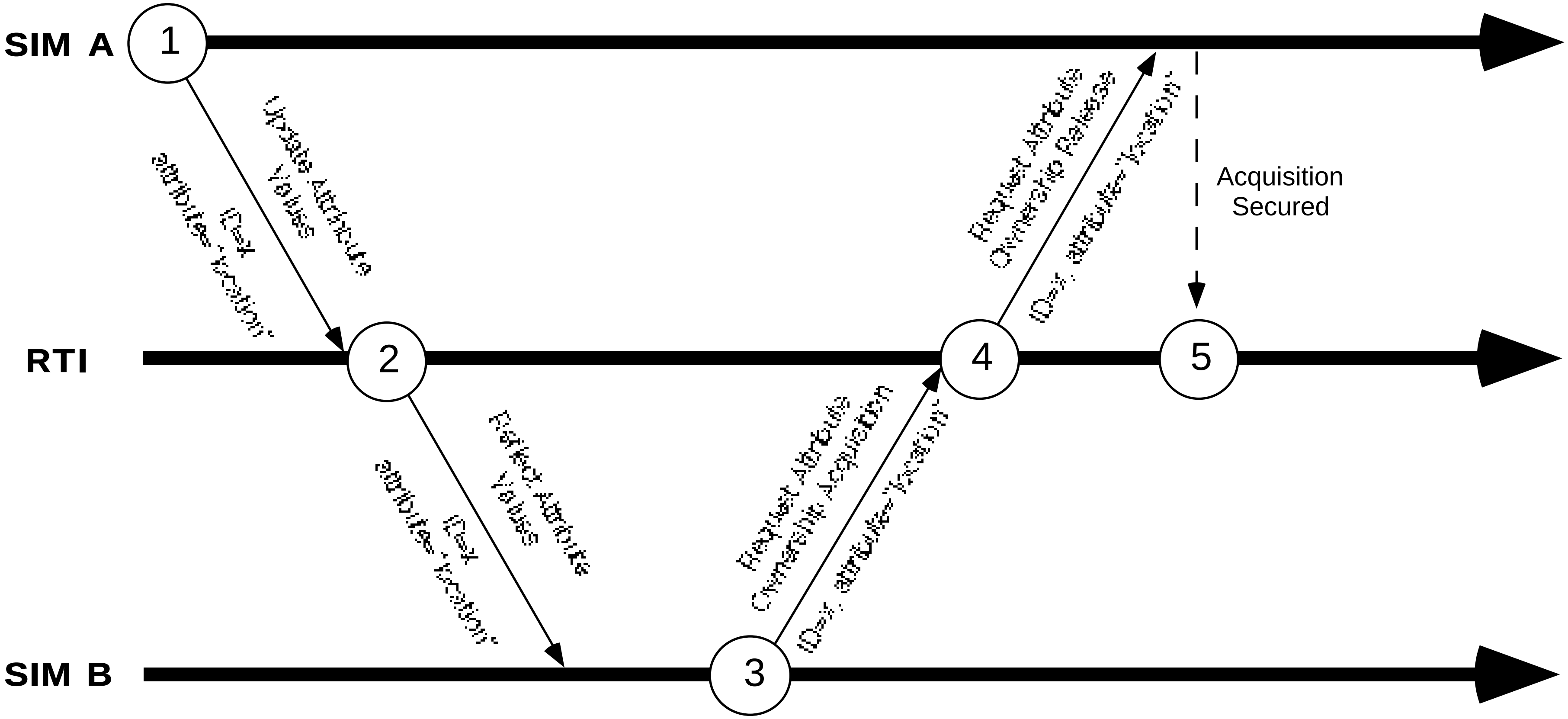
Duncan Miller
4/15/96

Ownership Management Services

Service	Invoked by
Request Attribute Ownership Divestiture	Simulation
Unconditional Attribute Ownership Divestiture	Simulation
Attribute Ownership Divestiture Notification	RTI
Request Attribute Ownership Assumption	RTI
Request Attribute Ownership Acquisition	Simulation
Attribute Ownership Acquisition Notification	RTI
Request Attribute Ownership Release	RTI
Query Attribute Ownership	Simulation
Request Delete Privilege Acquisition	Simulation
Request Delete Privilege Release	RTI
Delete Privilege Notification	RTI

Ownership Management Services

Acquiring Ownership

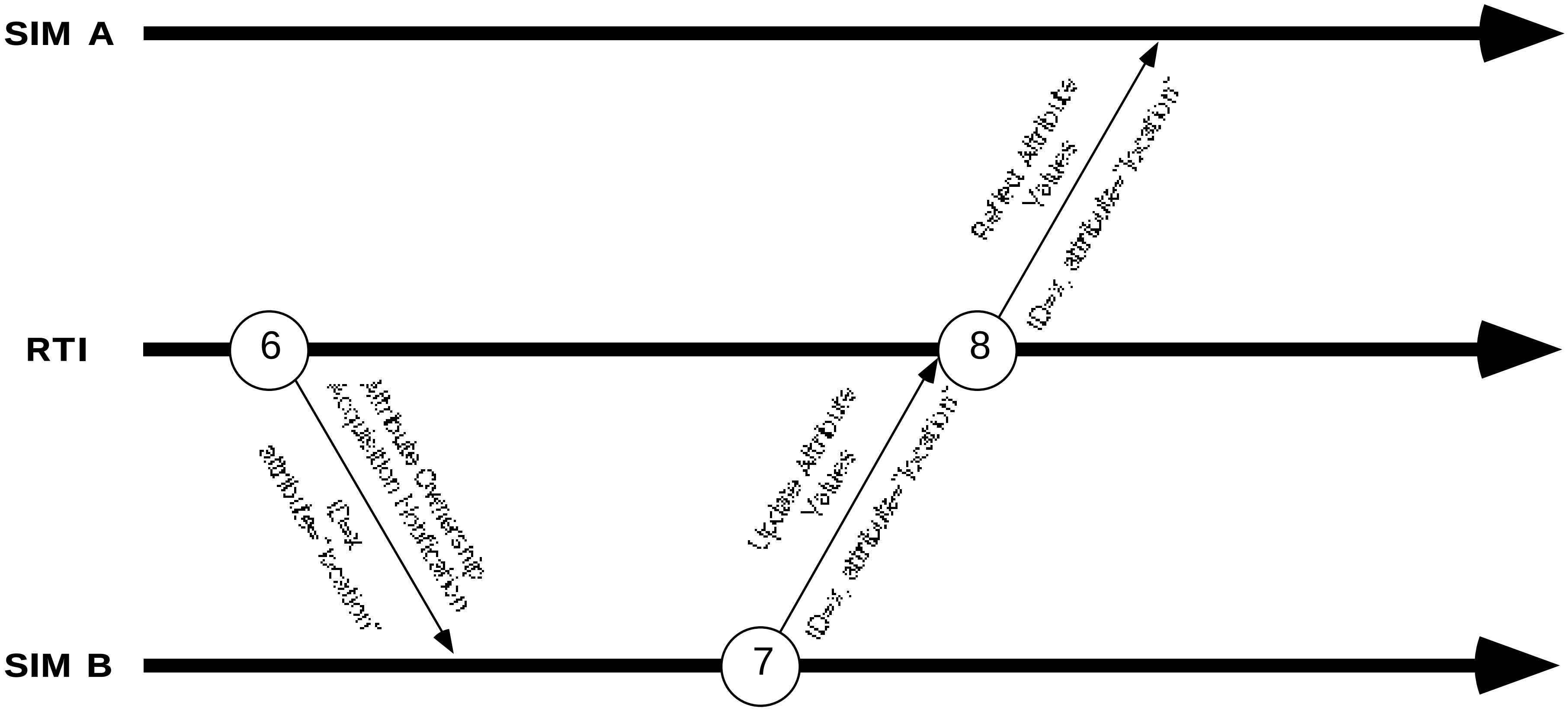

SIM A
RTI
SIM B
1
2
3
4
5
Update Attribute Values
ID=x attribute='location'
Reflect Attribute Values
ID=x attribute='location'
Request Attribute Ownership Acquisition
ID=x; attribute='location'
Request Attribute Ownership Release
ID=x; attribute='location'
Acquisition Secured

1. SIM A owns attribute "location" for object ID=x and is publishing updates
2. RTI is forwarding updates to SIM B
3. SIM B requests attribute ownership acquisition of attribute "location" for object ID=x
4. The RTI issues a Request for Attribute Ownership Release to the owner of object ID=x
5. The RTI receives the response that SIM A will release the ownership of the attribute

(continued ...)

Ownership Management Services

Acquiring Ownership (cont'd)

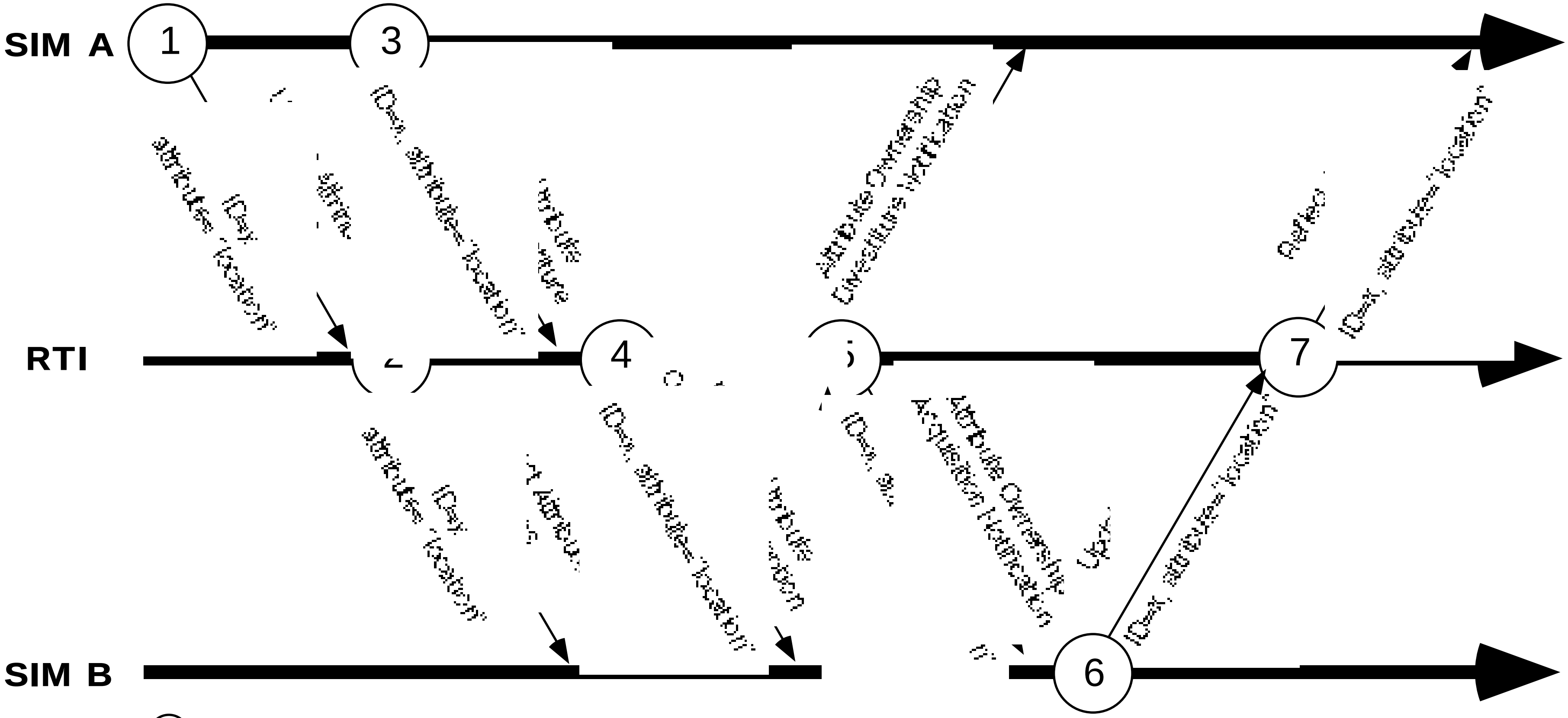

SIM A
RTI
SIM B
6
7
8
Attribute Ownership Acquisition Notification
(ID=x
attributes="location")
Update Attribute Values
(ID=x; attribute="location")
Reflect Attribute Values
(ID=x; attribute="location")

6 The RTI informs SIM B that SIM B acquired ownership of attribute "location" for object ID=x

7 & 8 SIM B begins producing updates for attribute "location" for object ID = x.
The RTI forwards these updates to subscribing simulators.

Ownership Management Services

Divesting Ownership


SIM A
RTI
SIM B
1
2
3
4
5
6
7
(ID=x, attribute="location")
Attribute Ownership Divestiture Notification
Attribute Ownership Acquisition Notification
(ID=x, attributes="location")

1. SIM A owns attribute "location" for object ID=x and is publishing updates
2. RTI is forwarding updates to SIM B
3. SIM A requests Attribute Ownership Divestiture of attribute "location" for object ID=x
4. The RTI issues a Request for Ownership Assumption to SIM B; SIM B accepts
5. The RTI receives the response that SIM B will accept ownership of the attribute and notifies SIM A that the attribute is released and SIM B to acquire the attribute

6 & 7 SIM B begins issuing updates for attribute "location" for object ID=x and the RTI forwards the updates

Example of IDL Representation

```
enum divestiture_status {
  RELEASED,
  RETAINED
};
typedef unsigned short simulation_id_type;
typedef unsigned long object_id_type;
typedef unsigned short attribute_name_type;
typedef sequence <attribute_name_type> attribute_name_set_type;
struct attribute_divestiture_status_pair_type {
  attribute_name_type attribute_name;
  divestiture_status status;
};
typedef
  sequence <attribute_divestiture_status_pair_type>
  attribute_divestiture_status_set_type;
[ ...]
interface federation_execution {
[ ...]
  void request_attribute_ownership_divestiture(in object_id_type          id,
                                               in attribute_name_set_type attrs,
                                               in simulation_id_type      sim);
  void unconditional_attribute_ownership_divestiture(in object_id_type          id,
                                                     in attribute_name_set_type attrs,
                                                     in simulation_id_type      sim);
  void request_attribute_ownership_acquisition(in object_id_type          id,
                                               in attribute_name_set_type attrs,
                                               in string                  reason);
  void query_attribute_ownership(in  object_id_type      id,
                                 in  attribute_name_type attr,
                                 out simulation_id_type  owner);
  void request_delete_privilege_acquisition(in object_id_type id,
                                            in string         reason);
};
```

Release Schedule

- Release Interface Definition Language (IDL) Specifications for all RTI services Aug 15
- Release C source code corresponding to IDL specs Aug 31
- Release Ada source code corresponding to IDL specs Sep 15

All releases will be accomplished by posting sources in a controlled area of the DMSO web server (www.dmso.mil)